Using Device Models to Facilitate the Retrieval of Multimedia Design Information

Catherine Baudin
Jody Gevins-Underwood
Recom Technologies, Inc.

Vinod Baya
Stanford University

Artificial Intelligence Research Branch
NASA Ames Research Center
Mail Stop 269-2
Moffett Field, CA 94035-1000

NASA Ames Research Center
Artificial Intelligence Research Branch
Technical Report FIA-93-21
July, 1993

Using Device Models to Facilitate the Retrieval of Multimedia Design Information

Catherine Baudin* **Jody Gevins Underwood****
AI Research Branch,
NASA Ames Research Center Mail Stop 269/2
Moffet Field, CA 94035 U.S.A.
baudin@ptolemy.arc.nasa.gov gevins@ptolemy.arc.nasa.gov

Vinod Baya
Center for Design Research
Bldg 530, Stanford University
Stanford, CA 94305, U.S.A.
baya@sunrise.stanford.edu

Abstract

An engineering team generates information in the form of meeting summaries, progress reports, engineering notes, spreadsheet calculations and CAD drawings. While complete "machine intelligible" models of an artifact design are still difficult to acquire, canned text information or videotapes of meetings are easy to capture but difficult to reuse. Dedal is an interface that facilitates the reuse of design experience by providing an intelligent guide for retrieving text, graphics and videotaped documents. It uses: (1) concepts from a model of the designed artifact to index and query design documents, and (2) a set of heuristics that reason from the model to "guess" where the answers to a question may be documented when the retrieval fails.

We observed an engineer asking questions to Dedal to access records of a shock absorber design and found that (1) using a model to index the records significantly increased the precision and recall of the information retrieved when compared with a baseline retrieval system, and (2) the heuristics contributed to this performance by increasing the recall of Dedal by 40%.

1. Motivation

To capture design information, researchers have investigated ways of acquiring both formal design records [Mostow et al. 87] [Baudin et al. 90] and informal text, graphic, and videotaped information [Lakin et al 89] [Stults 88]. While complete "machine intelligible" records of a design are still difficult to acquire in complex or innovative domains, canned-text or videotaped design information is easy to capture but difficult to retrieve because concepts in the records evolve over time and are closely interrelated.

Describing the content of design documents in terms of the artifact model facilitates information retrieval because: (1) it restricts the query vocabulary to the concepts in the model; and (2) when the retrieval fails, the relationships in the model can be used to predict where the answers to a query *might* be documented. This is based on the observation that pieces of information in a document are usually not isolated islands of ideas but are related by implicit rules.

Dedal is an interface to such records as meeting summaries, pages of a designer's notebook, technical reports, and videotaped conversations between designers. It takes a question from a user as input, matches the question against indexing patterns describing the *content* of the documents, and returns a list of references related to the question. If the retrieval fails, it applies a set of heuristics to "guess" where the answer to the query might be documented or where it could find an answer to a related question.

In this paper we focus on how Dedal uses an artifact model and a set of heuristics for information retrieval and present results from an experiment showing the contribution of the heuristics to the performance of the system. We found that the heuristics significantly increased the percentage of relevant references retrieved when compared (1) to the performance of the system without the heuristics, and (2) to a baseline boolean retrieval system based on string search. The heuristics however, decreased the precision of the system (number of relevant answers over the total answers retrieved) by a small but significant amount.

Sections 2 briefly presents the language used to index and query design records in the mechanical engineering domain. Section 3 presents the system's retrieval strategy. Section 4 describes examples of the heuristics used by Dedal. In Section 5 we report on our observation of an engineer asking questions to Dedal in the context of a design modification task.

* Employee of RECOM Technologies.
** Employee of Sterling Software, Inc.

To be published in the proceedings of the IJCAI 93 conference held in Chambery, France

This experiment is based on a study where we indexed documents generated for an innovative shock absorber designed for Ford Motor Corporation.

2. Background: A language to index and query design information

Engineering design documents describe aspects of an artifact design such as: (a) the *final product* : the structure and behavior of the device, its components, the interface between parts; (b) the *design process*: requirements, alternative options considered, rationale for choices, information sources. This information has different *levels of detail* , ranging from precise descriptions of a part to global views of an assembly, and can be conveyed through different *media* such as text, graphics, photos, tables, or equations.

Based on the information seeking behavior of designers conducted at Stanford's Center for Design Research and NASA Ames, we identified a language to describe and query design information [Baudin et al 92a]. This language combines concepts from a model of the designed artifact with a limited vocabulary representing generic task-dependent classes of information usually covered by design documents. For instances: "function," "operation," "alternatives" are elements of this vocabulary. Figure 1 shows the task vocabulary: the list of topics, level-of-details and media that describe the information in the records. The meaning of each task-dependent concept is described in a separate paper [Baya et al 92].

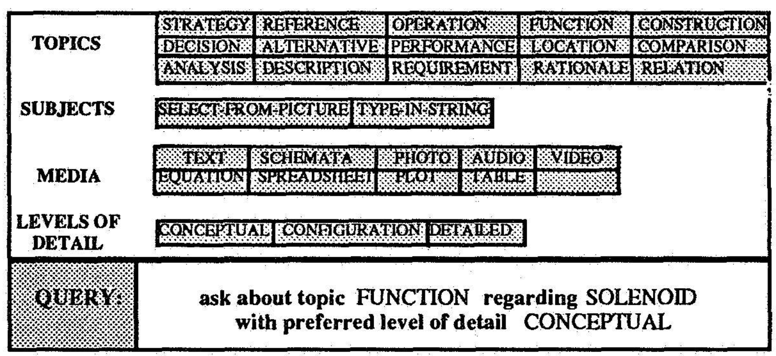

Figure 1: Query template

The domain model includes a representation of the artifact structure, some aspects of its function, the main decision points and qualitative relations derived from the equations used by the designers to model the device. It also includes concepts that are part of the problem but external to the device representation. The main relations in the domain model are *isa, part-of, attribute-of*, and *depends-on* (see Figure 2).

The information in a region of a record is described by one or several *indexing patterns* of the form: `<topic: T, subject: S, level-of-detail: L, medium: M>`. S is a list of subjects from the domain model. In addition, each indexing pattern contains a pointer to the record and segment corresponding to the starting location of the information. For instance: "The inner hub holds the steel friction disks and causes them to rotate" is part of a paragraph on page 20 of the record: report-333. It can be described by two indexing patterns:

- <topic: **function**, subject: **inner-hub**, level-of-detail: **configuration**, medium: **text**, in-record: **report-333**, segment: **20**>
- <topic: **relation**, subject: **inner-hub** and **steel-friction-disks**, level-of-detail: **configuration**, medium: **text**, in-record: **report-333**, segment: **20**>

The queries have the same structure as the indexing patterns and use the same vocabulary. Figure 1 shows a query template. A question such as: "How does the inner hub interact with the friction disks?" can be formulated in Dedal's language as:

- <topic: **relation**, subject: **inner-hub** and **steel-friction-disks**>

3 . Retrieval strategy

The retrieval module takes a query from the user as input, matches the question to the set of indexing patterns and returns an ordered list of references related to the question. The retrieval proceeds in two steps: (1) find the patterns that exactly match the query and return the associated list of references, (2) If the retrieval fails, activate a set of retrieval heuristics.

The selected references are then ordered following criteria such as: (1) the preferred default media: The assembly of a device might be better represented in the form of a visual sequence picturing how the device was assembled rather than by a detailed text description of the assembly; (2) the time stamp associated with a record: prefer the most recent records.

Each retrieval step returns a set of references. The user selects one and, if the document is on line[1], can go to the corresponding segment of information. A user dissatisfied with the proposed references can request

1 Dedal communicates with a system with hypertext access to its text and graphic documents. Audio documents are accessible on a Sparc workstation. Video is not yet directly accessible.

more information and force Dedal to resume its search and retrieve other references.

3.1 The need for using heuristics

There are two factors that make the retrieval of design information especially difficult:

(1) The concepts evolve over time. For instance, in our variable damper domain the concept: "force generation mechanism" evolved into the concept "actuator" and later into a precise component "solenoid". One implication of the dynamic nature of this information is that the answer to a question like "what are the reasons for choosing a solenoid rather than an electrical motor to generate a force?" might be discussed in a design meeting referring to the desired properties of the *actuator*, at a point in time where the concept of *solenoid* might not have yet been considered.

(2) Design concepts are closely interrelated. If a design parameter P1 depends on a design parameter P2, the reasons for choosing a value for P1 might be documented where the rationale for the value of P2 is described.

4. Using the domain model and the task vocabulary to find relations in design information

We analyzed different types of records such as engineering notebooks and structured progress reports. From this analysis we identified a set of twenty heuristics to help match a user's query with a set of indexing patterns that describe the documents. These heuristics do not depend on a particular design but on the record type, the task vocabulary and the type of relationships among the subjects in the domain model.

4.1 The retrieval heuristics

There are two different types of retrieval heuristics: the *proximity* heuristics look for information in the documents that are spatially related to the information required, and the *causal* heuristics look for information about concepts causally related to the subjects of a question. The current retrieval strategy is to activate the proximity heuristics before the causal heuristics.

(1) Looking for proximity relations: These heuristics look for regions in the documentation that contains information related to the query and assume that the required information will be found near these regions (on the same page, or in the same subsection depending on the type of record). For instance:

> equation-to-schemata: In an engineering notebook, an equation describing a mechanism will usually be found next to a schemata of this mechanism.

Some heuristics reason from the relations between the concepts in the task vocabulary:

> performance-to-analysis: Information about the *performance* of a particular assembly and the *analysis* of this assembly are likely to be located in nearby regions of the documentation.

Other rules exploit the hierarchical relations in the domain model:

> operation-to-function: In a structured document such as a progress report, the *function* of a part in a mechanical assembly might be found near where the *operation* of the assembly is described.

(2) Looking for causal relations: These heuristics reason about the dependencies among the attributes of a device. For instance:

> find-rationale: if a question is about *rationale* for the value of attribute Q1, and Q1 depends on Q2, then look for a segment of information describing the *rationale* for the value of Q2."

> find-a-common-ground: If a question is about a *relation* between Q1 and Q2, and Q1 depends on Q3, look for relations between Q2 and Q3.

4.2 Examples

Ex1: User question: "What is the function of the inner hub?"

Query in Dedal: `<ask-about function regarding inner-hub>`

Exact match: No information about *function* is associated with the subject *inner-hub*.

Heuristic selection and activation: "operation-to-function" is activated. "If the question is about *function of* X and X is part-of assembly Y, then look for patterns about *operation of* Y." (the definition of operation-to-function is shown in Figure 2,)[2] The component inner-hub is part-of the disk-stack assembly. The *operation-to-function* heuristic retrieves the indexing pattern:

```
<topic: operation, subject: disk-stack,
level-of-detail: configuration, medium:
text, in-record: spring-89-DRD-report,
segment: 20>
```

2 The $ symbols represent variables. in MRS [Genesereth 84], the first order predicate language, used to implement the rules.

The user selects this reference and sees the following statement in spring-89-DRD-report, page 20: "**The inner hub holds the steel friction disks and causes them to rotate** when the road input is transmitted through the connecting link to the rotating inner shaft."

Ex2: User question: "What is the rationale for the value of torque of damper?"

Query in Dedal: `<ask-about rationale regarding torque of damper>`

Exact match: There are no *rationale* topics associated with the attribute *torque of damper*.

Heuristic match: A proximity heuristic is applied. "If the question is *rationale* for X look for a pattern about *decision* for X". No information is associated with decision for torque of damper and the retrieval fails.

The causal heuristic "find-rationale" that looks for dependent attributes is activated (see figure 2). "If the question is about *rationale* for X and X depends on Y then look for patterns about the *rationale* for Y". The value of torque of damper depends on the value of resistive-force of disk-stack, the rationale for the value of resistive-force of disk-stack can be found in the record winter-90-DRD-report on page 15. The heuristic retrieves the indexing pattern:

```
<topic: rationale, subject: resistive-force of disk-stack, level-of-detail: configuration, medium: text, in-record: winter-90-DRD-report, segment: 15>
```

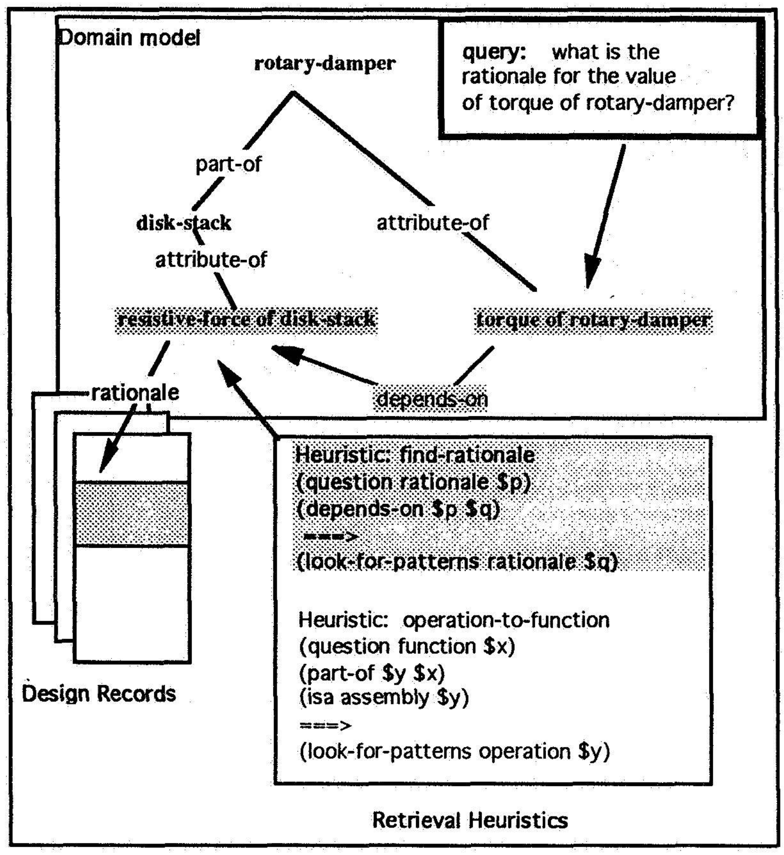

Figure 2: Using the domain model and heuristics to answer a query

5. Empirical Results and Analysis

We conducted an experiment where we focused on the ability of a user to formulate questions using Dedal, the ability of Dedal to retrieve relevant answers, and the global contribution of the inference engine to this performance.

5.1 The variable damper experiment

One of the designers of an electromechanical automobile shock absorber, the rotary damper, designed for Ford Motor Corporation, used Dedal to conceptually index the design documentation. Documents generated over a seven month period were put in the Electronic Design Notebook™ environment, a commercial hypertext system which communicates with Dedal. We conducted an experiment where we observed a novice mechanical engineer use Dedal to ask questions in the context of a modification of the shock absorber design [Baya et al. 92]. The subject then rated the answers as relevant or irrelevant. An answer was considered relevant (1) if the subject found the retrieved reference useful in the context of his question, or (2) if the damper designer who attended the experiment would have retrieved the same reference. For a given question more than one reference could be relevant.

5.2 Results

We selected 40 of the engineer's questions, whose answers were available in the documents, and submitted them to a baseline boolean retrieval system that searches for combinations of words in the text. We evaluated the retrieval performance of Dedal and the baseline system by measuring the *precision* (number of relevant references divided by the total number of references retrieved) and *recall* (number of relevant references retrieved divided by the total number of relevant references that are in the documents) [Blaire et al. 1989] of each system. These are two metrics commonly used to measure the performance of information retrieval systems. For example, 100% precision means that all references retrieved were relevant. 100% recall means that no relevant reference was missed.

Figure 3 shows the precision and recall of a baseline retrieval system and of Dedal with and without the activation of the heuristics. These results show that using concepts from the domain model to index and query the records greatly improve the precision of the retrieval (by 70 % in this study) when the heuristics are not activated. The heuristics improve the recall of Dedal's retrieval by 40% with a loss of 20% precision when compared to the results of Dedal using only an exact match.

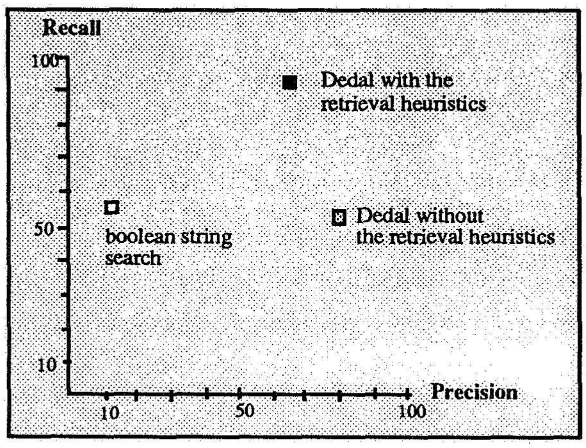

Figure 3: Information Retrieval Performances

5.3 Analysis

To improve the precision of the system when the heuristics are applied, we analyzed the factors responsible for the irrelevant answers retrieved by the inference mechanism. We considered four factors: (1) the weak heuristic factor where the heuristic used to retrieve a reference was too general or led to an incorrect inference; (2) the level of detail factor, where a question is more specific than the indexing patterns used to describe a document; (3) the bad indexing factor, where a document was improperly described; and (4) the query formulation factor, where a question was improperly formulated. We found that the main factors involved are the *level of detail* factor and the *weak heuristic* factor.

Level of detail factor: This factor covers the cases where the initial question asked is more specific than the vocabulary used to describe the document. For instance, the question: "Does the disk stack get pressed down?" is translated by the query: *operation of disk-stack*, as the notion of being pressed down is not part of the vocabulary. This means that some segments of information described by: *operation of disk-stack* might not answer the question while it exactly matches the query.

The choice of the task vocabulary and the domain vocabulary is a tradeoff between the simplicity of the indexing task and the precision of the querying procedure. If the indexing vocabulary is very detailed, the indexing task will be more arduous. If the indexing vocabulary is too abstract, the information retrieved will be imprecise. This factor is inherent to any modeling task in general and to any indexing scheme that attempt to model the semantics of the associated information in complex domains.

The weak heuristic factor: Most of the time this means that the heuristic used was too general in the context of a particular question. For instance, if a query is about the *operation of A* where A is a component, Dedal might try to match the query with a pattern *operation of B* where B is an assembly including A. As it appears after testing, the success of this heuristic depends on additional criteria such as: the complexity of the assembly B in terms of its number of parts and the size of the information segments in the documents. If the concept B has many parts and the whole document is about its operation, the chances of zooming in on the particular segment describing the operation of one of its subparts are slimmer than if B is an assembly that has few components and is described in only two pages of the document. We are in the process of refining our heuristics to take these kinds of specificity criteria into account.

To compensate for the loss in precision when the heuristics are applied we started to investigate the use of an *index refinement component* that caches the results of successful inferences. Each time Dedal uses a heuristic to find references, the user has the option to validate the references retrieved. Dedal then creates a new index: it combines the user query with the record and the segment associated with the relevant reference. In principle, this caching method should decrease the number of irrelevant references retrieved in response to a query because the next time a user asks the same question, the relevant answer will be retrieved by an exact match without activating the unsuccessful inferences. This method would be relatively straightforward if the weak heuristic factor was the only criterion involved in the retrieval of irrelevant answers by the inference mechanism. The problem is that because of the *level of detail* factor, two different questions can have the same representation in Dedal. Therefore an answer can be relevant one time and irrelevant for the same query another time.

6. Discussion

We use a conceptual indexing method based on a model of an artifact design to describe the content of multimedia design documents. The indexing patterns are structured objects whose meaning can be accessed by an inference mechanism to find relations among pieces of information in text, graphic or videotaped records. More experimentation is needed to evaluate Dedal, but our preliminary results suggest that mechanical engineers have been able to formulate questions in Dedal and that our model based indexing approach performs better than retrieval based on syntactic string search. However this type of conceptual indexing raises two concerns about the possible knowledge intensiveness of the approach: (1) it requires the definition of a domain model , and (2) the indices describe the *semantics* of a document and as such their acquisition is difficult to automate.

Domain dependence: The domain dependence of our approach may seems a burden compared to domain independent information retrieval methods. However, engineering design is a good area in which to try model-based conceptual indexing because: (1) a large part of the documentation can be described in terms of a structural model (*part-of* hierarchy) of the designed artifact. Unlike other device modeling aspects such as the prediction of a device behavior, or geometrical reasoning, device structure representation is well understood, and (2) the domain model may be used - or derived from models used - for purposes other than indexing, especially if the method is coupled with computer aided design features. For instance, an artifact model can be used to propagate changes of values in equations, to help diagnose faulty behavior or to validate the design against requirements. The fact that the domain model may have other purposes than indexing alleviates of the burden associated with the domain model definition.

Index acquisition: In Dedal, the indexing patterns are defined manually by a person knowledgeable in the domain such as one of the design engineers in the case of the rotary damper project. The heuristics facilitate the indexing task because the content of the document does not need to be exhaustively described, but the activation of the inferencing mechanism results in a loss in the precision of the information retrieved. To facilitate the index acquisition task without impacting retrieval performance, we started to investigate the use of incremental indexing techniques (see section 5.3), where the content of the documents is described at a high level of detail and the questions asked by designers are used to refine the document descriptions. The indexing method described in this paper facilitates the implementation of such an index refinement method because: (1) the query and the indices have the same structure, and (2) the indexing language and the retrieval heuristics enable high level indexing patterns to match more specific user queries.

7. Related Work

Using a model of the artifact to index design cases is a method investigated by the case-based reasoning community [Kolodner 84] [Sycara et al. 89] [Goel 89] [Birnbaum 89] for the purpose of retrieving a case by similarity with another. These lines of research have mainly been investigating how to generalize the representation of a case and its indexing to be able to find design similarities. This paper does not address the problem of case retrieval. We assume that the particular design case is selected by the user and that the problem is to access information from a case that is not completely formalized.

ASK Tom [Schank et al 91], a tutoring system developed at the Institute for the Learning Sciences, organizes multimedia information about a topic in a hypermedia network whose nodes are text and videoclips representing instructional cases and stories. Directly relating pieces of text manually to build a hypermedia network is a difficult task when there is a large amount of information. Our indexing method does not attempt to directly relate pieces of information. It first describes each segment of information in the records by relating them to a central domain model, which is then used to find relationships among the information in the documents.

Dedal uses semi-formal representations of design knowledge, as in the gIBIS argumentation-bases system [Conklin et al 88]. gIBIS uses *domain independent* concepts where the information is described in terms of *issues, arguments*, and *positions* to facilitate the capture of early deliberations. By contrast, our conceptual indexing scheme uses *domain dependent* and *task dependent* concepts to index and query multimedia documents. This enables the representation of the documents' content and the identification and use of retrieval heuristics to facilitate information access.

RUBRIC (Tong 89) uses natural language processing techniques and evidential reasoning to extract conceptual indices from a lexical analysis of textual documents. For instance, an evidential rule can define which words and relations among words suggest a given concept. These rules are domain dependent and must be redefined for each design problem. It is not clear at this point, how much background knowledge would be needed to automatically extract the record descriptions from our text-based records.

8. Conclusion

Our current set of retrieval heuristics is derived from documents generated at the Stanford Mechanical Engineering Department by several designers. More tests are needed to evaluate the generality of these rules and their ability to cover design documents generated by different organizations. However, we believe that even if the heuristics have to be refined or changed, the language and the concepts used to express these rules will remain the same. We are currently indexing the documents for a larger design project generated by a NASA design team.

In its present version, the indexing task is still a knowledge intensive activity that involves the participants of a design project. To scale up the approach we are now focusing on ways to: (1) integrate the acquisition of the domain model to the design process by coupling Dedal with CAD tools that use artifact models, and (2) to facilitate index acquisition by allowing indexing to proceed at a high level while using the queries to refine the document descriptions.

Acknowledgments

We are grateful to Fred Lakin from the Performing Graphics Company for his support of the Electronic Design Notebook which holds Dedal's text and graphics documents. We thank Ade Mabogunje, Larry Leifer and the other members of the GCDK group for their collaboration on this project. Thanks to Rich Keller, Phil Laird, Ian Underwood and Peter Friedland from NASA Ames and to Michel Baudin for their feedback on early drafts of the paper.

References

Baudin, C., Gevins, J, Baya, V, Mabogunje, A. 1992a "Dedal: Using Domain Concepts to Index Engineering Design Information", Proceedings of the Meeting of the Cognitive Science Society, Bloomington, Indiana.

Baudin, C., Gevins, J, Baya, V, Mabogunje, A. 1992b "Dedal: An Intelligent Interface for the Retrieval of Multimedia Design Information". NASA report FIA-TR-92-39.

Baudin, C., Sivard, C., Zweben, M., "Recovering rationale for design changes: A knowledge-based approach", in: Proceedings IEEE for Man, Machines and Cybernetics, Los Angeles, CA, (1990).

Baya, V, Gevins, J, Baudin, C, Mabogunje, A, Leifer, L. "An Experimental Study of Design Information Reuse", in proceedings of the 4th International Conference on Design Theory and Methodology, 1992.

Birnbaum, L. and Collins, G. , "Reminding and engineering design themes: a case study in indexing vocabulary" in Proceedings of the 1989 Workshop on Case-based Reasoning, edited by K. Hammond, Morgan Kaufman, San Mateo, CA, 1989.

Conklin, J., Begeman, M.L., "gIBIS: A Hypertext Tool for Exploratory Policy Discussion" 1988 ACM -0-89791-282-9/88/0140.

Genesereth, M, Greiner R., Grinberg M., Smith D., "MRS Dictionary", technical report Stanford University, Heuristic Programming Project, 1984.

Goel, A. and Chandrasekaran B., "Integrating Model-based Reasoning and Case-Based Reasoning for Design Problem Solving, " in Proceedings of the AAAI Design Workshop, 1989.

Kolodner, J. l., "Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model", Lawrence Wirlbaum Associates, Hillsdale, NJ, 1984.

Lakin F., Wambaugh J., Leifer L., Cannon D., and Sivard C., "The Electronic Design Notebook: Performing Medium and Processing Medium", THE VISUAL COMPUTER, International Journal of Computer Graphics" 1989 5: 214-226.

Mostow, J., and Barley M., "Automated Reuse of Design Plans", Proceedings of the International Conference on Engineering Design, February 1987.

Schank, R., Ferguson, W., Birnbaum, L., Barger, J., Greising, M., "ASK TOM: An Experimental Interface for Video Case Libraries" ILS technical report, March 1991.

Stults, R., "Experimental Uses of Videotapes to Support Design Activities", Xerox PARC Technical report SS1-89-19, December 1988.

Sycara K.P., and Navinchandra D. "Representing and Indexing Design Cases" in Proceedings of the Second International Conference on Industrial & Engineering Applications of AI and Expert Systems, Tullahoma, TN, June 1989, Vol. II, pp. 735-741.

Tong, M. R., Appelbaum, A., and Askman V. "A Knowledge Representation for Conceptual Information Retrieval", International Journal of Intelligent Systems. vol. 4, 259-283, 1989